PROBABILISTIC TERMINATION VERSUS FAIR TERMINATION

by

Michael Tiomkin

Technical Report #496
February 1988
Probabilistic Termination versus Fair Termination

Michael Tiomkin

Department of Computer Science,
Technion - Israel Institute of Technology, Haifa 32000, Israel

Abstract. In this paper we show that probabilistic termination of concurrent program is in many cases much simpler than the "fair" one. For a wide class of definitions of probabilistic termination we may express termination by $\Pi_2^0$ arithmetic formula, whereas the "fair" termination can be expressed only by $\Pi_1^1$ second order arithmetic formula. Proof of "fair" termination usually needs induction on recursive ordinals, but proof of probabilistic termination has the complexity equivalent to that of deterministic program termination.

1. Introduction

The notion of "fairness" (cf. [2]) naturally arises while dealing with verification of concurrent programs. It is generally accepted that for proving correctness of a concurrent program we may assume that the schedule has finite service time, i.e. if a process is enable long enough, it should be chosen. There exist many different notions of "fairness", e.g. the usual, the extreme and the absolute fairness, etc. As was pointed out by Hart, Sharir and Pnueli, all of these notions are based on different classes of possible deterministic schedulers (cf. [5]), or, equivalently, on different sets of infinite paths in the tree of all possible choices made by the schedulers. For every notion of "fairness", a program is said to be "fair" terminating, if the set of its infinite computations is disjoint with the set of "fair" paths.

One of the objections to the theories of "fair" computations is their high complexity. For example, "fair" termination of a recursive concurrent program is $\Pi_1^1$ complete, and proving such termination involves induction on countable recursive ordinals, whereas termination of a recursive deterministic program may be expressed by a $\Pi_2^0$ arithmetic formula.

Alternatively, many papers appeared in the last years consider probabilistic schedules (cf. [5], [8], [11]). That is to say, instead of the class of admissible deterministic schedules for the "fairness", one can
easily define the class of admissible probabilistic schedules, and instead of one fixed set of infinite "fair" paths one can deal with sets of paths which have probability 1 for every "admissible" probabilistic schedule. We shall say that a program terminates almost everywhere (a.e.), if the set of its finite computations has a measure 1 for every "admissible" probabilistic schedule. Then one can easily prove that "fair" termination implies a.e. termination for different sets of schedules. For example, if the only admissible schedule chooses the possible processes with equal probabilities, then "equifair" termination (and almost all other fair terminations) implies a.e. termination (cf. [4]). If we allow schedules choosing any possible process with probability greater than $\epsilon$, for some $\epsilon > 0$, then "fair" ("extremely fair") termination implies a.e. termination (cf. [8], [11]).

The reason for dealing with probabilistic schedulers is as follows. First, in the real system the "closed world" assumption is not exactly true. Any real scheduler (operating system) depends on the events from the external world, such as the speed of channels and processors, the input data from measuring devices, and, the worst, human interface. This makes the probabilistic analysis of system behavior so attractive. Second, after years of extensive (and successful!) using pseudo-random numbers, one might feel that a compound digital-analogue machine with the real random number generation might be much more convenient. Clearly, we can not promise the exact probability distribution we need (recall that here we deal with analogue device). We may be only assured that the distribution is "close" enough. This brings us to the notion of "stable" probability distribution (cf. [11]), where the probability of choosing a process must be "far enough" from 0.

However, many computer scientists still prefer discrete versions of program semantics and verification. The purpose of this paper is to show that, in the case of verification of concurrent programs, the non-discrete reasoning is much simpler than the discrete one. We shall see that for "simply definable" classes of probabilities a.e. termination is $\Pi_2^0$. Thus a.e. termination of a concurrent program and termination of a deterministic program are of the same complexity.

In detail, this paper is organized as follows. In Section 2 we give some basic definitions and notation. Section 3 contains the main results and a short discussion dealing with sets of probabilities having higher degree of definability. Section 4 contains the acknowledgement to Michael Kaminski (Technion).
for fruitful discussions and critical reading the paper.

2. Definitions and notation

We introduce here the habitual definitions of a computation tree for finite number of concurrent sequential processes, or, equivalently, finitely nondeterministic program. In addition, we define some (topo)logical and probabilistic notions which will be used in the sequel.

For a finitely nondeterministic program \( \xi \) (cf. the definition of the Guarded Commands language, \( GC \), in [1] or [2]) with the fixed input data, we use a standard definition of its computation tree \( T_5(\xi) \) (cf. [3]). We can easily extend this Guarded Commands language with the usual iteration constructs, such as \( \text{while} \) or \( \text{repeat} \), avoiding the "trivial" nodes of out-degree 1 and making \( T_5(\xi) \) a recursively enumerable \( (RE) \) tree instead of the primitive recursive one. This is similar to what is done in biology, economics and sociology, treating differently the stable and the unstable stages of the process (cf. [7]). All the results remain unchanged in this case. This tree has a finite out-degree (bounded by the maximal number of guards in the selection statements). Moreover, for a node \( \sigma \) with out-degree greater than 0, the exact out-degree of \( \sigma \) can be recursively computed. Obviously, the tree \( T_5(\xi) \) with the out-degree \( \leq n \) can be recursively mapped to an \( RE \) subtree of a full \( n \)-tree, \( T_n = \{1, \ldots, n\}^\infty \).

A computation of a program \( \xi \) is a maximal path in the tree \( T_5(\xi) \). There exists a natural topology on the set \( T_5^{\text{max}}(\xi) \) of all maximal paths on \( T_5(\xi) \). This topology is generated by the clopen sets \( U_\sigma = \{ p \in T_5^{\text{max}}(\xi) : p \text{ passes through } \sigma \} \), where \( \sigma \) is a node of \( T_5(\xi) \). For a program \( \xi \), the set of infinite computations of \( \xi \), i.e., the set of maximal infinite paths in the corresponding \( T_5(\xi) \), is closed in this topology. The set of \( \xi \)'s terminating states \( \text{Term}_\xi \) is an \( RE \) set of nodes of \( T_5(\xi) \). Further on we shall deal with the open set \( \text{TERM}_\xi = \bigcup_{\sigma \in \text{Term}_\xi} U_\sigma \subseteq T_5^{\text{max}}(\xi) \) generated by the set of terminating states of \( \xi \). All these \( U_\sigma \) are mutually disjoint one-element clopen sets. Notice that the set \( \text{TERM}_\xi \) is not always complement to the set of infinite paths of \( \xi \); there may be states where \( \xi \) loops deterministically.

We give also some definitions concerning probabilities on the Borel subsets of \( T_5^{\text{max}}(\xi) \).
We define probability as a nonnegative measure which is equal to 1 on the whole space. \(Pr(T_s^{\text{max}}(\xi))\) is the set of probabilities on the Borel subsets of \(T_s^{\text{max}}(\xi)\). It is well known that every measure on Borel subsets of topological space is uniquely determined by its values on the base open sets. In our case, every probability from \(Pr(T_s^{\text{max}}(\xi))\) is uniquely determined by its values on the sets \(U_\sigma\). These values, in their turn, are uniquely determined by the conditional probabilities to pass from a node of \(T_s(\xi)\) to its immediate successor. Actually, the above conditional probabilities are defined on \(E(T_s(\xi))\), the set of edges of \(T_s(\xi)\). Sometimes a conditional probability is not defined (the probability to arrive to \(C_1\) is 0). In this case we can assume it to be equal to any value from the segment \([0,1]\), providing that the sum of the probabilities to exit a node is equal to 1.

So every probability from \(Pr(T_s^{\text{max}}(\xi))\) may be naturally mapped into the space \(M_{S_\xi} = \{0,1\}^{E(T_s(\xi))}\).

The set \(M_{S_\xi}\) with the product topology (generated by the finite products) is a compact (as the product of compacts). The set of probabilities is defined by the demand that for every node \(\sigma\) the sum of conditional probabilities on edges exiting \(\sigma\) is equal to 1. Thus the set of probabilities is a closed subset of \(M_{S_\xi}\), hence it is a compact as well. In the next section we only use the compactness of \(Pr(T_s^{\text{max}}(\xi))\) and the trivial fact that the value of a probability on \(U_\sigma\) is a continuous function on \(M_{S_\xi}\) (it depends continuously on conditional probabilities on fixed finite number of edges from \(E(T_s(\xi))\)).

Now this is the time for some logic. We define the notion of a.e. termination, and introduce definable sets of probabilities from \(Pr(T_s^{\text{max}}(\xi))\).

**Definition 2.1.** For a fixed set of probabilities \(SP \subseteq Pr(T_s^{\text{max}}(\xi))\), we shall say that a program \(\xi\) terminates \(SP\) almost everywhere (a.e.), if for every probability \(p \in SP\) \(p(\text{TERM}_\xi) = 1\).

One can find examples of \(SP\) a.e. termination in [11] (the "pseudoprobabilistic" termination), or [8] (the "P-valid" termination). Since we are interested here in the definability properties of different notions of termination, we shall deal only with sets of probabilities which are definable in some theory.

**Definition 2.2.** We shall say that the set \(SP \subseteq M_{S_\xi}\) has closed recursive algebraic definition, or, shortly, is a \(RAF\) set, if

1) \(SP\) is closed.
2) For every finite set of edges \( Ed \subseteq E(T_5(\xi)) \) the projection of \( SP \) on \( Ed \), i.e. the set

\[
\{ f \in [0,1]^{Ed} : f \text{ may be extended to some } p \in SP \}
\]

is definable by a formula \( \varphi_{Ed}(x_1, \ldots, x_{|Ed|}) \) in real analysis (theory of real numbers with operators "+" and "\cdot", predicates "=" and "<", and variables and quantifiers over reals, cf. [10]). Notice that by compactness of \( MS_5 \) and 1) this projection is a closed subset of \([0,1]^{Ed}\).

3) The formula \( \varphi_{Ed} \) can be obtained recursively from \( Ed \) (notice that a finite \( Ed \subseteq E(T_5(\xi)) \) is a constructive object).

Remark that a closed set \( SP \) may be easily reconstructed from its projections, it is equal to the intersection of the cylinders raised from its projections.

The following proposition shows that the set of probabilities is a well-defined subset of \( MS_5 \).

**Proposition 2.3.** The set \( Pr(T_5^{\max}(\xi)) \) is a RAF set.

**Proof.** For a finite set of edges \( Ed \subseteq E(T_5(\xi)) \) we construct a formula defining the projection of \( Pr(T_5^{\max}(\xi)) \) on \( Ed \). We define the set of the left nodes of \( Ed \) as

\[
Le(Ed) = \{ \sigma \in T_5(\xi) : (\sigma, \sigma') \in Ed \text{ for some } \sigma' \}.
\]

For a node \( \sigma \in Le(Ed) \) the out-degree \( d_\sigma \) of \( \sigma \) in \( T_5(\xi) \) is greater than 0, hence this degree may be obtained recursively from \( \xi \) and \( \sigma \). For a node \( \sigma \in Le(Ed) \) we define also the nonempty set of edges

\[
Ed_\sigma =\{ e \in Ed : e = (\sigma, \sigma') \text{ for some } \sigma' \} \subseteq Ed.
\]

We can define the projection of \( Pr(T_5^{\max}(\xi)) \) on the set \( Ed_\sigma \) by the formula

\[
\psi_\sigma = \begin{cases} 
\sum_{e \in Ed_\sigma} p(e) \leq 1, & \text{if } |Ed_\sigma| < d_\sigma, \\
\sum_{e \in Ed_\sigma} p(e) = 1, & \text{if } |Ed_\sigma| = d_\sigma.
\end{cases}
\]

Then the projection of \( Pr(T_5^{\max}(\xi)) \) on the set \( Ed \) is defined by the formula \( \& \sigma \in Le(Ed) \psi_\sigma \). This formula depends recursively on program \( \xi \) and set \( Ed \). ■

Because intersection of two RAF sets is a RAF set, Proposition 2.3 implies the following corollary.

**Corollary 2.4.** If a set \( SP \subseteq MS_5 \) is a RAF set, then the set of probabilities from \( SP \), i.e. the set \( SP \cap Pr(T_5^{\max}(\xi)) \), is also a RAF set.

Recall that because of the finite encoding of RAF sets, one can easily define countable unions and
intersections of RAF sets, etc.

Definition 2.5. We shall say that a set \( SP \subseteq M_\delta \) has a recursive algebraic \( F_\sigma \) definition, or, shortly, is a RAE\( _\sigma \) set, if \( SP = \bigcup SP_n \), where each \( SP_n \) is a RAF set, and the formula \( \varphi_\delta^{SP} \) that defines projection of \( SP_n \) on a finite set of edges \( Ed \subseteq E(T_\delta(\xi)) \) can be obtained recursively from \( Ed \) and \( n \).

E.g., one can see that the set of "stable" probabilities (cf. [11]), i.e. probabilities with \( p(e) \geq p > 0 \), for every \( e \in E(T_\delta(\xi)) \), this set is a RAE\( _\sigma \) set. The set consisting of one probability with recursively computable algebraic values is trivially a RAF set.

In the next section, we shall deal with expressibility of a.e. termination for RAE\( _\sigma \) sets of probabilities.

3. The Main Results

We are going to prove that a.e. termination for a RAF\( _\sigma \) set of probabilities may be expressed by a \( \Pi_2^0 \) formula. For this we shall use a well-known fact from Mathematical Analysis:

Fact 3.1. If \( \{f_n\} \) is a sequence of continuous real functions on a compact \( K \), and \( \{f_n\} \) converges pointwise monotonically to a continuous function \( f \) on \( K \), then the sequence \( \{f_n\} \) converges to \( f \) uniformly on \( K \).

For the proof cf. [9, Th. 7.13, p. 150].

From this fact we obtain the following corollary.

Corollary 3.2. Let \( SP \subseteq Pr(T_\delta^{\max}(\xi)) \) be a RAF set of probabilities. Then the SP a.e. terminating of a program \( \xi \) with fixed input data can be expressed by a \( \Pi_2^0 \) formula. This formula depends recursively on \( SP \) and \( \xi \).

Proof. The set of terminating states of \( \xi \) is RE. Let \( term_k \) be the set of terminating states computed by the enumerating process after \( k \) steps of computation. We have, trivially, that \( term_k \subseteq term_{k+1} \subseteq Term_\xi \), and \( Term_\xi = \bigcup_k term_k \). The value of \( term_k \) depends recursively on \( k \) and \( \xi \). Now, for an integer \( k \) we define a function \( f_k \) on \( Pr(T_\delta^{\max}(\xi)) \) by

\[
f_k(p) = p(\bigcup_{\xi \notin term_k} U_\alpha) = \sum_{\xi \notin term_k} p(U_\alpha),
\]

for a probability \( p \in Pr(T_\delta^{\max}(\xi)) \). The sequence of functions \( \{f_k\} \) converges pointwise monotonically to
p(TERM\textsubscript{\xi}). Also, the RAF set \( SP \) is compact as a closed subset of \( Pr(T_{S}^{max}(\xi)) \). Hence if \( \xi \) is \( SP \) a.e. terminating, i.e. the functions \( \{f_{k}\} \) converge pointwise to 1 on \( SP \), then, by Fact 3.1, they converge to 1 uniformly. Thus we can express the \( SP \) a.e. terminating of \( \xi \) by the expression

\[
\forall m \exists k \forall p \in SP \ p(\bigcup_{\text{o.e. term}} U_{\alpha}) > 1 - \frac{1}{m}.
\]

Notice that the probability \( p(\bigcup_{\text{o.e. term}} U_{\alpha}) \) depends only on a finite number of values of \( p \) on \( E(T_{S}(\xi)) \). Projections of \( SP \) on finite number of edges on \( E(T_{S}(\xi)) \) are expressible in real analysis. Thus, because the validity of a formula from real analysis is recursive (cf. [10]), we can recursively test the validity of the formula \( \forall p \in SP \ p(\bigcup_{\text{o.e. term}} U_{\alpha}) > 1 - \frac{1}{m} \) for every given pair \( m,k \).

Now, using Corollary 3.2 one can easily prove the same result for RAF\textsubscript{a} sets of probabilities and programs with variable input data.

**Theorem 3.3.** For a RAF\textsubscript{a} set of probabilities \( SP \subset Pr(T_{S}^{max}(\xi)) \), and a program \( \xi \) with RE set of finite possible input data, \( SP \) a.e. terminating of a program \( \xi \) is expressible by a \( \Pi_{2}^{0} \) formula.

**Proof.** Let \( SP \) be a RAF\textsubscript{a} subset of \( Pr(T_{S}^{max}(\xi)) \). By Definition 2.5, \( SP \) is equal to \( \bigcup_{\xi} SP_{k} \), where \( SP_{k} \) is a RAF subset of \( Pr(T_{S}^{max}(\xi)) \). Then the following two conditions are equivalent.

1) for every \( p \in SP \), \( p(TERM_{\xi})=1 \).

2) for every \( k \), for every \( p \in SP_{k} \), \( p(TERM_{\xi})=1 \).

Therefore we can express the \( SP \) a.e. termination of \( \xi \) by the following statement:

"for every possible input \( Inp \), and for every integer \( k \), \( \xi(Inp) \) is \( SP_{k} \) a.e. terminating".

By Corollary 3.2, this expression is \( \Pi_{2}^{0} \) (it is a \( \Pi_{2}^{0} \) expression with 2 additional universal quantifiers from the outside).

In many cases this theorem allows us to prove that probabilistic termination is \( \Pi_{2}^{0} \). E.g., we say that a probability \( p \) is *stable* if there exists \( \varepsilon > 0 \), such that \( p(e) > \varepsilon \) for all \( e \in E(T_{S}(\xi)) \) (cf. [11]). We define that a program \( \xi \) terminates "pseudoprobabilistically" (cf. [11]) if the set \( TERM_{\xi} \) has probability 1 for all stable probabilities. Then we have the following corollary.

**Corollary 3.4.** The "pseudoprobabilistic" termination is \( \Pi_{2}^{0} \).

**Proof.** The "pseudoprobabilistic" termination is the a.e. termination for the set of stable probabilities. The
set of "stable" probabilities is $\text{RAF}_\alpha$, and the result follows from Theorem 3.3.

As it was pointed out in Section 2, one can easily define recursive algebraic Borel subsets of $\text{Pr}(\text{Tr}(\bar{a}))$ of any finite (and even infinite recursive) degree. We can define "open" recursive algebraic set (or, shortly, $\text{RAG}$ set) as a complement to a $\text{RAF}$ set, and, similarly, $\text{RAF}_\alpha$ and $\text{RAG}_\beta$ sets etc. One can easily see that every $\text{RAG}$ set is also a $\text{RAF}_\alpha$ set, hence Theorem 3.3 holds for $\text{RAG}$ sets as well.

Unfortunately, the method used in the proof of Theorem 3.3 is not applicable for the sets of the degree higher than $\text{RAF}_\alpha$. In this connection, we would conjecture the following.

Conjecture 3.5. There exists a $\text{RAF}_\alpha (\text{RAG}_\beta)$ subset $\text{SP}$ of $\text{Pr}(\text{Tr}^{\text{max}}(\bar{a}))$, such that the SP a.e. termination of a program is $\Pi^1_1$ complete.

4. Acknowledgement

The author is very grateful to M. Kaminski for fruitful discussions and critical reading the paper.

References


1965).


